

# ROS Basics

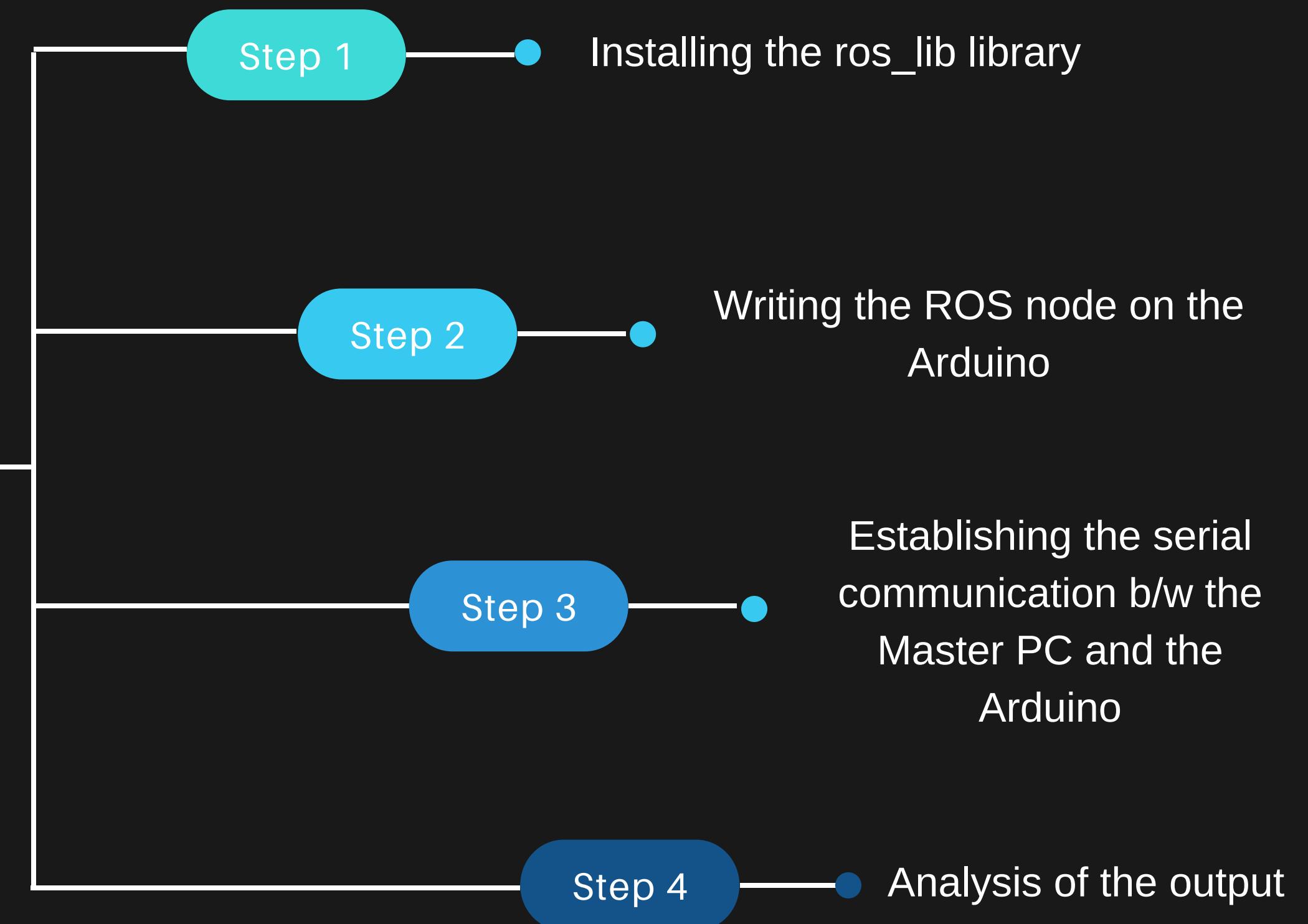
**Topic Covered:** ROS integration with Arduino

Part 2

100Days of ROS series

By  
M.N.Subhash  
@malladisubhash

# RUNNING ROS ON ARDUINO



## Step 1: ROS\_LIB

- The Communication b/w the ROS and the Arduino is possible through the rosserial\_arduino package
- rosserial provides the ROS communication protocol that works over the Arduino UART
- This package allows Arduino to act a full fledged ROS node and publish and subscribe the ROS message directly
- To use this rosserial\_arduino package a arduino library is developed which is ros\_lib which needs to be added to *libraries* folder of the Arduino IDE

**Source :** [http://wiki.ros.org/rosserial\\_arduino/Tutorials/Arduino%20IDE%20Setup](http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino%20IDE%20Setup).

## Step 2: Writing the programs to make arduino act as publisher and subscriber node

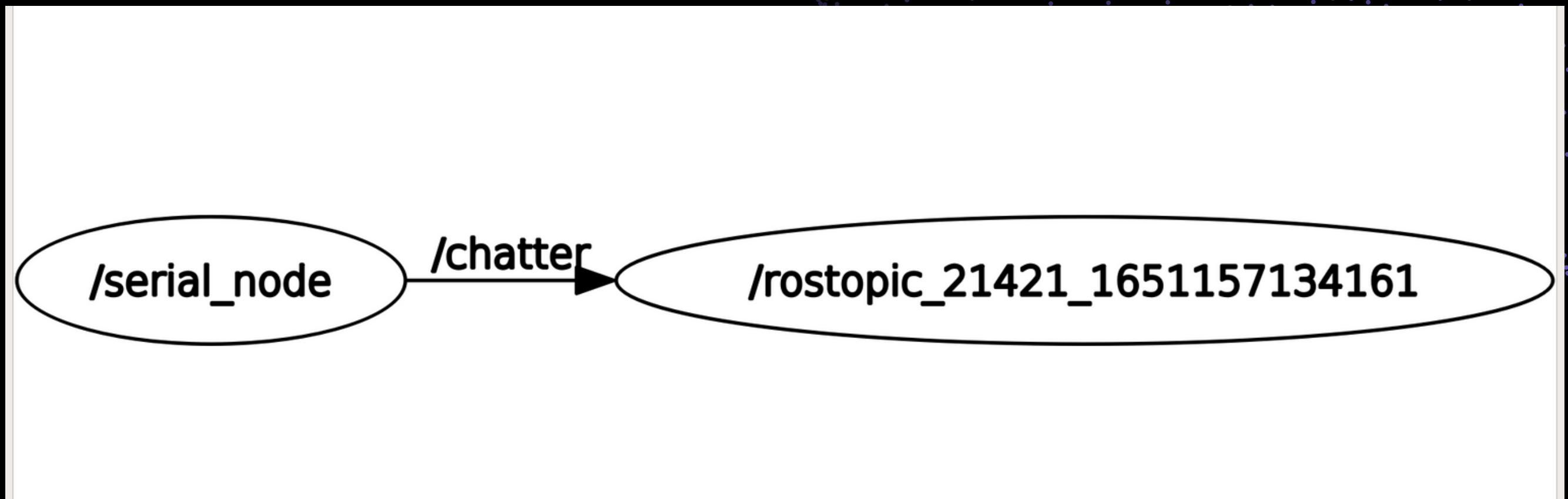
- As the part of the challenge I have worked on two examples in the first, I made arduino to act as a publisher node to publish hello world message
- In the second example I have worked on the blinking led example where the arduino acts as the subscriber node
- The detailed analysis of the code will be done in the next part of the series

**Source :** [http://wiki.ros.org/rosserial\\_arduino/Tutorials>Hello%20World](http://wiki.ros.org/rosserial_arduino/Tutorials>Hello%20World) ,  
[http://wiki.ros.org/rosserial\\_arduino/Tutorials/Blink](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink)

## Step 3 & 4: Establishing the Communication, Running the Node and analysing the o/p

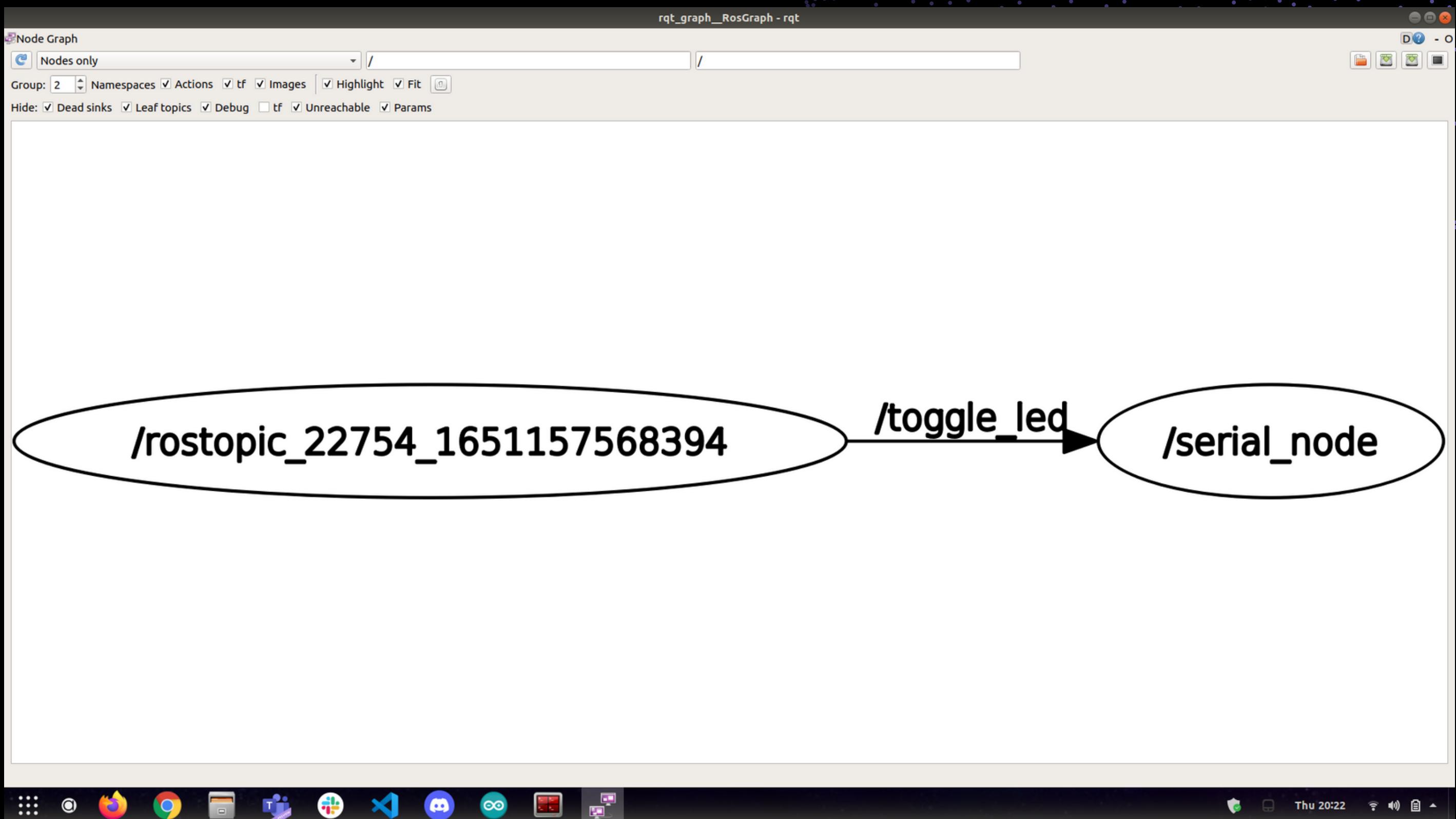
- The communication b/w the ROS master pc and Arduino is established by the UART port.
- The process is as follows
  - **Run the command :** `rosrun rosserial_python serial_node.py <port_number>`
  - This command runs the `serial_node.py` file in the `rosserial_python` package, to establish the connection b/w the arduino and master PC
  - Once the connection is established all the libraries and the rosmessages which are used in the node that is running on the arduino are verified and the arduino starts publishing/ subscribing to the data
- Now to see the output of the task 2:
  - As in this task arduino is the publisher and the master PC is the subscriber, we can listen to the message that is being published by the arduino to the channel **/chatter**
  - Now, to listen run the following code: `rostopic echo /chatter`
  - The main learning of this task are the integration of ROS with arduino and analysing rosmessages of the type **/std\_msg/string** which is one of the common message type

- The rqt graph of the task is as follows:



- Now, in task 3, The arduino acts as a subscriber and the Master PC acts as publisher
  - Once the communication b/w the both is established by running the same commands
  - By running the command : rostopic pub toggle\_led std\_msgs/Empty --once
  - This command publishes an empty message to the topic **/toggle\_led** topic which accepts the message type **std\_msgs/Empty**, now the arduino which has subscribed to the topic will listen to the message and toggle the LED on.
  - The main learning of this topic are controlling the LED interfaced to arduino using ROS and understanding another message type of the same **std\_msgs** family which publishes empty messages

- The rqt graph is as follows



# About Me

I am currently pursuing my degree at Sir M Visvesvaraya Institute of Technology, Bangalore.

The domains I am interested in are Electronics and Robotics.

My interest in Robotics started when I was in my second year, Currently improving my skills in ROS and took a 100daysofROS challenge to focus mainly on the hardware implementation of ROS

I put regular updates of my progression of challenges on Twitter  
As a part of the challenge, I will be documenting the important things and will be making a concise version of them.

Follow me to catch up with ROS and learn how to implement it on hardware.

**LinkedIn profile:** [linkedin.com/in/malladisubhash](https://linkedin.com/in/malladisubhash)

**Twitter Handle:** [@mns2610](https://twitter.com/@mns2610)

**GitHub ROS repository link:** [https://github.com/malladi2610/100\\_days\\_of\\_ROS](https://github.com/malladi2610/100_days_of_ROS)